

## APLIKACJA DO PRZETWARZANIA OBRAZÓW MEDYCZNYCH Z WIZUALIZACJĄ PRZESTRZENNĄ

Andrzej KRZYŻANOWSKI<sup>1</sup>, Paweł POCZEKAJŁO<sup>2</sup>

1. Politechnika Koszalińska, Wydział Elektroniki i Informatyki  
tel.: 665-281-830 e-mail: shaidel@wp.pl
2. Politechnika Koszalińska, Wydział Elektroniki i Informatyki  
tel.: 94 3478 721 e-mail: pawel.poczekajlo@tu.koszalin.pl

**Streszczenie:** W pracy przedstawiono opracowanie prototypu aplikacji umożliwiającej przegląd i przetwarzanie trójwymiarowych (3D) obrazów medycznych. Aplikacja umożliwia płynne wyświetlanie przekrojów 2D obiektu wzdłuż każdej z osi (x,y,z) oraz za pomocą algorytmu renderingu wolumetrycznego pozwala na wyświetlenie obiektu w 3D. Dodatkowo poprzez wykorzystanie efektu odbicia na ścianach bryły o kształcie ściętego ostrosłupa aplikacja umożliwia uzyskanie obrazu 3D, jako wizualizacji przestrzennej. Praca zawiera dokładny opis algorytmu generującego tekstury 2D i 3D pozwalające na wyświetlenie obiektu, omówienie funkcjonalności wraz z instrukcją użytkowania, a także na przykładzie wykorzystania zestawu deweloperskiego Jetson Nano 2GB propozycję konfiguracji zestawu prototypowego do wyświetlania obrazów medycznych. W podsumowaniu pracy omówiono pomysł wykorzystania aplikacji w celach edukacyjnych oraz możliwości dalszego jej rozwoju.

**Słowa kluczowe:** rendering wolumetryczny, obrazowanie medyczne, Python, OpenGL

### 1. WSTĘP

Analiza sygnałów biomedycznych, zwłaszcza obrazów, stanowi bardzo ważny aspekt współczesnej medycyny. Informacje, jakie niosą ze sobą dane obrazowe, są używane zarówno w celach diagnostycznych, jak też przy doborze terapii. Obrazowanie medyczne obejmuje wiele zagadnień, wśród których należy wymienić rejestrację obrazów, przetwarzanie (np. poprawę jakości, segmentację), analizę (wydobywanie określonych parametrów), jak i śledzenie ruchu na sekwencjach obrazów. Obrazy medyczne dostarczają zazwyczaj informacji o anatomicznej strukturze narządów, ale mogą też służyć do wizualizacji ich fizjologii.

Najczęściej spotyka się obrazy dwu- i trójwymiarowe przedstawiające statyczne obiekty. Można także analizować obrazy dynamiczne, zmieniające się w czasie, a więc trój- lub czterowymiarowe. Często surowy obraz otrzymywany na wyjściu urządzenia obrazującego nie może być wykorzystany przez lekarza i nie pozwala na postawienie poprawnej diagnozy. Koniecznym narzędziem jest system pomiarowy z istotnym udziałem przetwarzania danych przez komputer, bez którego diagnostyka medyczna nie mogłaby istnieć.

#### 1.1. Cel pracy

Celem pracy jest opracowanie prototypu urządzenia umożliwiającego przegląd i przetwarzanie obrazów medycznych oraz szczegółowe omówienie algorytmu

renderingu wolumetrycznego, który pozwala na wygenerowanie z serii zdjęć 2D obrazu we wszystkich osiach układu x-y-z i obrazu 3D danego obiektu. Aby tego dokonać przygotowano oprogramowanie, a następnie przedstawiono propozycję konfiguracji zestawu rozwojowego Jetson Nano 2 GB wraz z peryferiami. Głównym przeznaczeniem urządzenia HoloMed jest wykorzystanie w celach edukacyjnych. Może posłużyć jako pomoc dydaktyczna w nauce inżynierii biomedycznej, przetwarzania obrazu oraz w nauce anatomii, jako oprogramowanie prezentujące wybrane fragmenty ludzkiego organizmu.

### 2. STANDARD DICOM [1]

DICOM - obrazowanie cyfrowe i wymiana obrazów w medycynie (ang. Digital Imaging and Communications in Medicine) to norma opracowana przez ACR (American College of Radiology) i NEMA (ang. National Electrical Manufacturers Association) dla potrzeb ujednoczenia wymiany i interpretacji danych medycznych reprezentujących lub związanych z obrazami diagnostycznymi w medycynie. DICOM znajduje zastosowanie głównie w przetwarzaniu obrazów tomografii komputerowej (TK), obrazowania metodą rezonansu magnetycznego (RM), pozytonowej tomografii emisyjnej (PET), cyfrowej angiografii subtrakcyjnej (DSA), cyfrowej radiografii konwencjonalnej (CR), radiografii cyfrowej (DR) oraz wszystkich wykorzystujących technologie cyfrowych badań o wysokiej rozdzielczości obrazu. Dane w formacie DICOM mają dużą objętość, wymagają specjalnego oprogramowania i sprzętu komputerowego, a także łączą o wysokiej przepustowości, za to pozwalają zachować wysoką jakość obrazu. Stosowanie normy DICOM umożliwia, między innymi, funkcjonowanie teleradiologii (umożliwienie równoczesnej pracy nad tymi samymi danymi poprzez udostępnianie ich w sieci).

#### 2.1. Tagi DICOM [2]

Każdy plik DICOM przedstawia dane w postaci sekwencji elementów zawierających informacje o obrazie, przy czym ostatni element ma sam obraz. Każdy element składa się z kilku części, w tym:

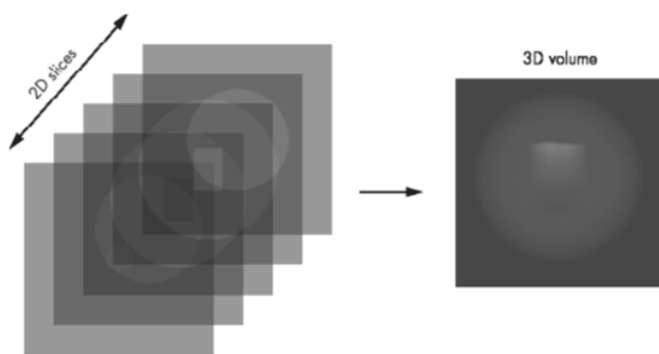
- tagów (ang. Tag) – unikalnych identyfikatorów elementów składających się z pary wartości reprezentowanych

- w systemie szesnastkowym i określających numer grupy i numer elementu,
- typów danych (ang. Value Representation) – linii przedstawiającej skrót typu danych,
- rozmiaru danych (ang. Value Length),
- informacji ang. (Data) – informacji zawartych w elemencie, w tym sam obraz.

Jeżeli pole danych (ang. Data) zawiera informacje, których nie można przedstawić w postaci tekstu (na przykład obraz lub wykres), to odpowiednia kolumna będzie miała postać: *Binary content.Hidden* (zawartość binarna, ukryta). Jeśli Tag jest standardowy, zostanie wyświetlona jego nazwa. Jeżeli Tag jest zarezerwowany przez producenta danej modalności lub w jego nazwie jest błąd, to w polu pojawi się: *Unknown Tag & Data* (nieznany Tag i dane).

### 3. ALGORYTM RENDERINGU WOLUMETRYCZNEGO [3]

Obrazowanie za pomocą rezonansu magnetycznego i tomografii komputerowej jest procesem diagnostycznym tworzącym dane wolumetryczne, które składają się z obrazów 2D pokazujących przekroje poprzez wolumen 3D. Rendering objętościowy, zwany też wolumetrycznym rekonstruuje obrazy 3D na podstawie zbioru danych wolumetrycznych. Dane MRI i CT mają zazwyczaj postać siatki 3D w formie  $N_x * N_y * N_z$  lub wycinków 2D w formie  $N_z$ -obrazów, gdzie każdy wycinek jest obrazem o rozmiarach  $N_x * N_y$  [4]. Dane wycinków wyświetlane są jako przezroczyste, co poprzez nałożenie na siebie kolejnych warstw tworzy obraz 3D (przykład Rys.1). W aplikacji zastosowano metodę renderingu wolumetrycznego opartego na obrazach, wykorzystywaną do generowania finalnego obrazu z wycinka 2D piksel po pikselu.

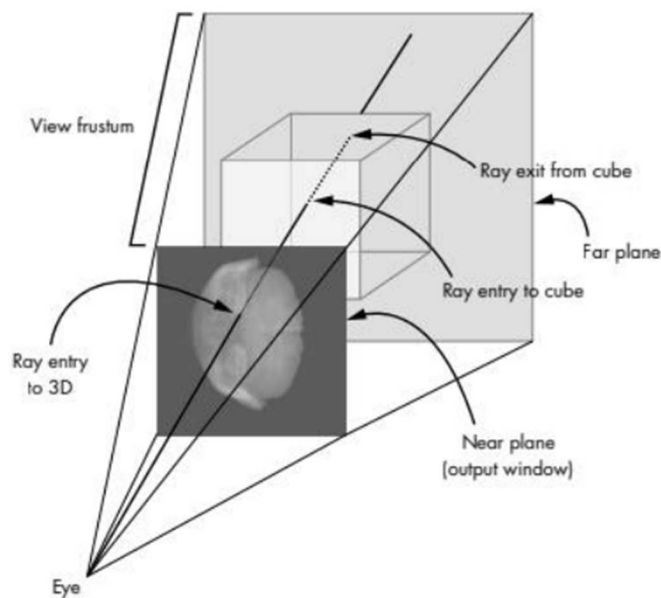


Rys. 1. Obraz 3D jako efekt nałożenia na siebie kolejnych warstw

#### 3.1. Generowanie promieni

Zastosowana metoda renderingu wyprowadza promień dla każdego piksela obrazu wyjściowego w kierunku dyskretnego zbioru danych wolumetrycznych 3D. Gdy promień przechodzi przez wolumen, dane są próbkowane, a próbki łączone w celu obliczenia wartości koloru lub intensywności końcowego obrazu (przykład na rys. 2).

Zgodnie z rysunkiem 2 celem jest wygenerowanie rzutu perspektywicznego danych wolumetrycznych 3D. Rysunek prezentuje obiekt OpenGL i pokazuje w jaki sposób promień wyprowadzony z oka wchodzi do bryły widzenia w bliższej płaszczyźnie, przechodzi przez wolumen danych i wychodzi przez dalszą płaszczyznę.

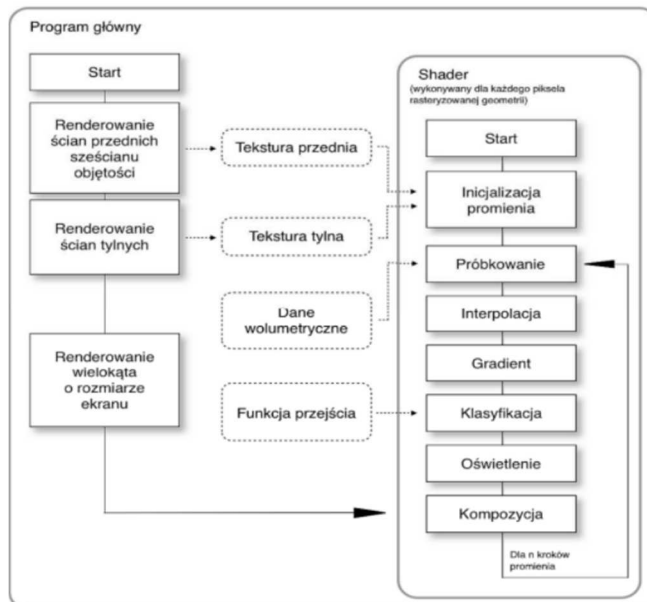


Rys. 2. Schemat powstawania obrazu 3D, jako rzutu perspektywicznego

#### 3.2. Generowanie tekstury 3D

Teksturę 3D uzyskano poprzez wykorzystanie klasy *Image* biblioteki *PIL* do wczytania obrazu do 8-bitowej tablicy *numpy*. Następnie upewniono się, że wszystkie obrazy ze zbioru mają tę samą wysokość i szerokość. Po załadowaniu indywidualnych tablic tworzona jest finalna tablica 3D powstająca dzięki połączeniu tych tablic z wykorzystaniem metody *concatenate* z biblioteki *numpy*.

#### 3.3. Schemat blokowy zastosowanego algorytmu [5]



Rys. 3. Schemat blokowy algorytmu renderingu wolumetrycznego

### 4. PROPOZYCJA KONFIGURACJI ZESTAWU PROTOTYPOWEGO

Prototyp urządzenia HoloMed został oparty o minikomputer Nvidia Jetson Nano w wersji z 2 GB pamięci RAM. Za wyświetlanie obrazu odpowiada ekran Waweshare LCD 13,3 cala o rozdzielczości 1920x1080 pikseli. Całości dopełniają bezprzewodowe mysz i klawiatura.

Aby uzyskać obraz przestrzenny przygotowano bryłę o kształcie ostrosłupa ze ściętym wierzchołkiem, na którego ścianach odbijany jest obraz obiektu 3D, co przedstawiono na rysunku 4.



Rys. 4. Umieszczenie bryły do wyświetlania obrazów przestrzennych na ekranie

#### 4.1. Oprogramowanie

W celu umożliwienia prezentacji obrazów medycznych, zarówno jako wycinki 2D, jak i obiekt 3D prezentowany na ekranie lub jako obraz przestrzenny przygotowano program, który poprzez rendering i przetwarzanie danych wejściowych pozwala na wyświetlanie wyżej wymienionych obrazów.

Katalog z oprogramowaniem zawiera:

- aplikację główną *volrender.py*,
- aplikację pomocniczą, służącą do konwersji obiektów zapisanych w standardzie DICOM do plików graficznych TIFF DCM\_ (*Img.py*),
- aplikacje do tworzenia danych testowych *makedata.py* (dane zapisywane są w katalogu *sphere-cuboid*),
- trzy katalogi ze zbiorami skanów medycznych [6].

#### 4.2. Funkcjonalności

Program główny HoloMed, uruchamiany za pomocą pliku *volrender.py* (przykłady działania przedstawiają rys. 5 i rys. 6) umożliwia:

1. Wyświetlanie obrazów medycznych 2D.
2. Wyświetlanie obiektów 3D powstałych przy wykorzystaniu algorytmu renderingu wolumetrycznego.
3. Wyświetlanie obrazów 2D w rzucie z dowolnej osi x-y-z, jako efektu transformacji zbioru danych wejściowych.
4. Uzyskanie obrazu quasi-3D na ścianach fizycznej bryły o kształcie ściętego ostrosłupa (poprzez efekt odbicia obrazu z wyświetlacza).
5. Płynne przeglądanie kolejnych obrazów 2D.
6. Płynne obracanie obrazu 3D.
7. Przybliżenie i oddalenie prezentowanego obiektu 3D.

Program pomocniczy DCM\_to\_Img.py umożliwia:

1. Odczyt informacji o parametrach graficznych danych zawartych w plikach DICOM.
2. Zapis danych graficznych plików DICOM, jako pliki TIFF.

Aplikacja do tworzenia danych testowych *makedata.py* umożliwia:

1. Wygenerowanie zbioru danych testowych.
2. Sprawdzenie poprawności działania aplikacji głównej.



Rys. 5. Przykładowy render obiektu 3D

#### 4.3. Wymagania

Podstawowe wymagania i funkcjonalności systemu:

1. Program musi być aplikacją wieloplatformową, działać na komputerach z systemami operacyjnymi rodziny Windows, Linux i MacOS.
2. Program powinien być aplikacją lokalną (desktopową).
3. Wykorzystane w programie biblioteki powinny być darmowe i otwarte-źródłowe (ang. open source).
4. Aplikacja powinna być możliwa do uruchomienia na komputerze z 2 GB RAM.

#### 4.4. Realizacja

HoloMed został wykonany jako aplikacja lokalna, napisana w całości w języku programowania Python, co dzięki wieloplatformowości tego języka pozwoliło na zrealizowanie pierwszego z wymogów nieformalnych. Równie ważnym powodem wyboru tego języka programowania była możliwość wykorzystania bibliotek *Pydicom*, *Numpy* i *Pillow*, które z jednej strony umożliwiły konwersję plików DICOM na pliki graficzne, a z drugiej łatwe przetwarzanie obrazów i transformację macierzy będących podstawą tekstur. Nie bez znaczenia jest też fakt, że wszystkie trzy wymienione biblioteki są bibliotekami *open source* czym spełniają trzeci wymóg nieformalny. Szybkość działania aplikacji jest zależna od wielkości zbioru danych wejściowych, najwolniej uruchamia się przy największym zbiorze (katalog *ct\_woman\_corpse*). Nie mniej nawet tu pozwala na płynne przetwarzanie danych i wyświetlanie zarówno obrazów 2D, jak i 3D na komputerze z jedynie 2 GB pamięci RAM.

#### 5. PODSUMOWANIE

Przygotowany prototyp zgodnie z założeniem pozwala na prezentację i analizę skanów medycznych, zarówno jako obrazów 2D, jak i 3D. Spełnione zostały zaplanowane funkcjonalności, a przebieg testów potwierdził prawidłowe działanie HoloMedu. Możliwości pozyskania danych medycznych z otwartych repozytoriów oraz wykorzystanie aplikacji pomocniczej do konwersji danych DICOM na pliki graficzne pozwala na wyświetlanie obrazowania medycznego wykonanego w dowolnej technologii, począwszy od USG oraz tomografii komputerowej i na rezonansie magnetycznym kończąc.



Rys. 6. Obraz przestrzenny obiektu *ct\_woman\_corpse*

Proponując wykorzystanie mikrokomputera Nvidia Jetson Nano w wersji 2 GB RAM postarano się także o stosunkowo niski koszt prototypu. Najdroższym elementem okazał się ekran, który jednak by zapewnić pełne przedstawienie obrazu przestrzennego powinien być dużo większy niż zaproponowany w artykule ekran 13,3 cala. Pewnym ograniczeniem jest też wymóg zaciemnienia pomieszczenia, w którym chcemy uzyskać odbicie obrazu na ścianach bryły stożka oraz konieczność prezentacji obiektu na wysokości wzroku obserwatora. Pomijając te ograniczenia prototyp spełnił pokładane w nim nadzieje, dostarczając nowatorskiego rozwiązania w dziedzinie wyświetlania skanów medycznych.

Zadowalający wynik prac nad prototypem pozwala myśleć o dalszym rozwoju aplikacji. Możliwe kierunki rozwoju to:

- dodanie segmentacji obrazów, pozwalające na wyraźniejsze ukazanie uszkodzeń, urazów lub zmian patologicznych,
- wykorzystanie sieci neuronowych w celu rozpoznawania wzorców, a przez to wspomaganie pracy chirurga poprzez ukazanie miejsc w organizmie pacjenta, które mogą być obszarami zmian chorobowych.

## 6. BIBLIOGRAFIA

1. Norma PN-EN ISO 12052:2017-10 „Informatyka w ochronie zdrowia – Obrazowanie cyfrowe i przesyłanie obrazów w medycynie (DICOM), łącznie z przepływem zadań i zarządzaniem danymi”
2. <https://www.dicomlibrary.com/dicom/dicom-tags/> (dostęp z dnia 26 września 2022)
3. Manesh Venkitachalam, “Python Playground: Geeky Projects for the Curious Programmer”, No Starch Press 2016
4. John Vince, „Mathematics for Computer Graphics”, Wydawnictwo Springer 2017
5. Kruger J., Westermann R., Acceleration Techniques for GPU-Based Volume Rendering, IEEE Visualization 2003
6. <https://graphics.stanford.edu/data/voldata/> (dostęp z dnia 26 września 2022)

## APPLICATION FOR PROCESSING MEDICAL IMAGES WITH SPATIAL VISUALIZATION

The article presents the development of a prototype of an application enabling the review and processing of medical images. The application enables the smooth display of 2D sections of an object along each axis ( $x$ ,  $y$ ,  $z$ ) and, using the volumetric rendering algorithm, it enables the object to be displayed in 3D. In addition, by using the reflection effect on the walls of a solid in the shape of a truncated pyramid, the application allows you to obtain a 3D image as a 3D visualization. The work contains a detailed description of the algorithm that generates 2D and 3D textures that allow displaying the object, discusses the functionality with the instructions for use, and uses the Jetson Nano 2GB development kit as an example, a proposal to configure a prototype set for displaying medical images. The summary of the work presents the idea of using the application for educational purposes and the possibility of its further development.

**Keywords:** volumetric rendering, medical imaging, Python, OpenGL.